ICOT Technical Report: TR-0795

TR-0795

GRAPE: Knowledge acquisition support groupware for the classification-choice problem

by
H. Ueda & S. Kunifuji (Fujitsu)

August, 1992

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome

(03)3456-3191~5
Telex ICOT J32964
Minato-ku Tokyo 108 Japan

Institute for New Generation Computer Technology

GRAPE: Knowledge acquisition support groupware for the classification-choice problem

— The design principles, groupware functions, and a suggestion to extend for the planning problem —

Haruyasu Ueda [1] Susumu Kunifuji [2]

[1] International Institute for Advanced Study of Social Information Science, FUJITSU Lab. LIMITED., 17-25, Shinkamata 1-Chome, Ota-ku, Tokyo 144, Japan

[2] Department of Computer Science, Japan Advanced Institute of Science and Technology, HOKURIKU., Asahidai 15, Tatsunokuchi, Ishikawa 923-12, Japan

Abstract

We have developed a knowledge acquisition support groupware GRAPE (GRoupware for Acquiring, Processing, and Evaluating knowledge). GRAPE helps to acquire the necessary knowledge for the rapid prototyping system for the classification-choice type system from scratch. In other words, GRAPE acquires all the names of the objects, all the structures and all the evaluation values necessary of computing the results. To acquire such a variety of knowledge, it incorporates various methods well-known in the system engineering area.

GRAPE is not only a knowledge acquisition support tool but also groupware. Groupware means that GRAPE has the features for the cooperative work support. This support includes the WYSIWIS (What You See Is What I See)[1] interface and the knowledge merge mechanisms. The WYSIWIS interface provides interface in which multiple users can input to GRAPE at the same time, and can see shared knowledge immediately. This interface gives each user the feeling that they are cooperating each other. It also stimulates the idea of the respective user by showing the idea inputed by other users.

The main mechanisms for groupware are merging every knowledge from each expert while keeping validity. This merged knowledge becomes shared knowledge. Generally, the simple average or concatenation is used as a mechanism for merging acquired knowledge, except for structure knowledge. Structure knowledge, like tree structure, is at once transformed into averagable forms like a matrix. The knowledge in averagable form is averaged and then the average is transformed back as the shared knowledge.

We restricted GRAPE to acquire knowledge for classification-choice type system, because of the simplicity. We will show a possible extension to acquire knowledge for the planning type system. To extend GRAPE to acquire knowledge for planning system, we designed a logical language named GCL[2]. GCL solves various constraints and generates

solutions according to evaluation function, then the generated solutions can be applied to present GRAPE.

This paper shows design principles, methods used to support acquisition, methods used to merge knowledge, and demonstrations of GRAPE. Then, it shows an extension for the planning type system of GRAPE using GCL.

1 Introduction

Knowledge acquisition is most important for the knowledge-based system, because the knowledge acquisition cost often takes half or more than half the time of development of a knowledge-based system. There are a lot of systems available to support the knowledge acquisition[3].

The phase making rapid prototype system is most important when we construct a knowledge-based system. A large knowledge-based system needs so much knowledge and the ill structured design makes the knowledge more complex than the knowledge needed for a well structured design. The ill structured design is always expensive, to complete and to modify the system, and sometimes it makes it impossible to complete the system. The rapid prototype system makes it easy to verify the design of knowledge, which is called initial knowledge.

There are different kinds of knowledge acquired by using knowledge acquisition tools, for example rules, decision tree and so on[4, 5, 6, 7]. But there is many another kind of knowledge especially needed as initial knowledge. It includes the structure of objects, the structure of attributes and the evaluation method of solutions. These small but various and complete sets of knowledge need to be acquired to make a rapid prototype system.

When a knowledge engineer makes a rapid prototype, (s)he tries to simulate the expert about some small and simple examples. Then (s)he finds how many kinds of knowledge are necessary for the knowledge-based system. The data for the small and simple examples are usable as the initial knowledge. After the initial knowledge is acquired, the knowledge engineer simply adds knowledge to the initial knowledge. In other words, (s)he does not need to add any other kind of knowldge to the initial knowledge.

GRAPE is intended to be a replacement of the knowledge engineer. It asks the users how to simulate to make results, then produces initial knowledge.

GRAPE is also groupware. This means GRAPE acquires knowledge from multiple users, and it has special features to manage multiple users. There exists no other knowledge acquisition tool which has the groupware features as far as we have researched.

The groupware has the following advantages as a knowledge acquisition support tool. To cooperate to input the knowledge, each user reduces the amount of input. Additionally, the knowledge from the multiple users tends to be less biased and more complete than a single user. It is because the missing of the knowledge is completed by the knowledge from other users. It is also because the knowledge inputed by other users stimulates a user and the user tends to input more knowledge than the knowledge amount of a user inputs alone. This is the same effect as what is often said "Two heads are better than one." This effect makes it possible to acquire knowledge from skill-less experts, who are less expensive than skillful experts. This effect also makes possible to acquire knowledge from non experts, who are deep concern with the knowledge based system, for example the manager of the development of a knowledge-based system and the end users of the knowledge-based system. In this paper, we will not use the phrase "human expert." Instead of this phrase, we will use the word "participant" to emphasize that GRAPE is groupware and acquires knowledge from multiple users including non experts.

To acquire knowledge for the planning type system, the special function is needed. It is because there are huge numbers of solutions. To find some candidates needs so much computation if the candidates are generated in a simple way. We will show an extension using language GCL designed to generate solutions for planning type system. It is designed to solve the constraints, which is described using inequality, non-equal relation, and linear inequality, and to produce the only solutions satisfying these constraints. It also handles the non-linear evaluation function, and search the solutions using the best-first search method.

In the following section, we will show the design principles of GRAPE including the outline of GRAPE, then the detail of the methods used in GRAPE including some examples, then an extension of GRAPE for planning problem using the constraint logical language named GCL, and finally conclusion.

2 Design of GRAPE

As mentioned in the previous section, we intend to develop a replacement system for the knowledge engineer, which supports acquiring initial knowledge for a rapid prototype system. In this section, first we will define the domain of the knowledge-based system for which GRAPE acquires. Then we will show the design principles for the knowledge acquisition tool and the groupware incorporated in GRAPE. Last, we will show the outline of the sequence of the knowledge acquisition.

2.1 The domain of the knowledge to acquire

There are many types of knowledge-based systems, for example, the planning type system, the diagnosis type system, classification-choice type system, and so on. We limited the type of the knowledge-based system to the classification-choice type system. The classification-choice type system is a system to chose one solution from the possible candidates of the solutions, and the candidates are acquired from participants. This limitation makes the problem simple but still quite difficult, because this type of knowledge-based system concentrates on the most important thing, which is how to evaluate and decide the solution. This type of system also has advantage that it needs less knowledge than other types of systems, and there are several techniques to choose the solution well-known as system engineering methods. Even the classification-choice type systems need various types of knowledge. The following knowledge is necessary at a minimum: the names of the candidates, the names of the attributes, the importance among attributes, the evaluation values of the candidates with each attribute. The knowledge tends to be so big that a structure of knowledge will be needed; a structure for the candidates and a structure for the attributes. GRAPE acquires these six types of knowledge.

The technique used for the classification-choice types of systems can be used any other type of knowledge-based system, because any kind of system must produce solutions and chose one from them. However, in a system for the planning, the problem of how to generate the candidates is as difficult as how to select the solution from the candidates. It is because there are huge numbers of solutions and to find some candidates needs so much computation if the candidates are generated in a simple way. We will show an extension of GRAPE to acquire the knowledge for planning type system in section 5.

2.2 Principle as knowledge acquisition support system

GRAPE acquires knowledge by letting participants actually choose one solution from the candidates using the system engineering techniques. This is a similar method to the way that the knowldge engineer acquires the initial knowledge for the rapid prototype system of a classification-choice type system. This method has three advantages. First, this method is simple and easy to input. Any user unfamiliar with knowledge acquisition can contribute to the initial knowledge as participants. Second, there are some system engineering methods to help the participants choose one solution from the candidates. The system engineering method helps find missing or duplicated knowledge, which includes the candidates and the evaluation values. It makes the initial knowledge better. Third, the participants actually choose the solution and agree with the result. This means the knowledge used to choose the solution is tested once. The participants can choose another solution from the same knowledge as many times as they want. If the participants do not agree with the solution, they will modify the knowledge until GRAPE produces the satisfiable solution. So, the quality of knowledge is rather good than knowledge without a test.

We used the following system engineering techniques: PCP (Personal Construct Psychology)[8], Extended ISM (Interpretive Structural Modeling)[9], and AHP (Analytic Hierarchy Process)[10]. PCP is used to acquire the names of the attributes to evaluate the candidates. Extended ISM is used to describe and to arrange the attributes, and AHP is used to elicit the importance among the attributes and the evaluation values among the candidates. PCP and AHP are known well as knowledge acquisition methods as well as the system engineering methods[4, 6]. Extended ISM is extended to describe the complicated relations between attributes. The details of the methods are described in section 3.

We selected these methods according to following principles:

- The knowledge acquisition course can be divided into several small courses, and one system engineering method can be used for each small course.
- The sequence of the knowledge acquisition course can be fixed and enough knowledge can be acquired without backtracking.
- Each system engineering method has the facility to keep the knowledge consistent.
- Related to following subsection, the knowldge acquired from the participants can be merged into one consistent knowledge.

2.3 Principle as groupware

GRAPE also has groupware aspects. This means that GRAPE acquires the knowledge from multiple users, and it has special features to manage the participants. The knowledge acquisition groupware has the three following advantages:

- Each participant can reduce the amount of input.
- Each participant can complete the unsure knowledge from the knowledge from the other participants.
- The knowledge inputed by a participant stimulates another user and the stimulated participant tends to input more knowledge than when the participant inputs alone.

In spite of these advantages, there is no knowledge acquisition groupware because it is difficult to make a knowledge acquisition tool groupware.

The main reason for this difficulty is that the participants must wait for other participants. Especially when the acquisition of the knowledge is based on other acquired knowledge. In that case, the system needs to require all participants to agree with the completeness of the preceding knowledge before proceeding to the next knowledge. The waiting participants feels tired and this decreases their efficiency. So, the design of the agreement point is important. We designed GRAPE with as small an agreement point as possible.

There are other difficulties in the knowledge acquisition groupware.

- The best methods to merge the knowledge acquired from participants may not be trivial.
- In case there are no appropriate merging methods, the participants must discuss and negotiate how to make a shared knowledge.
- How should we regard respective participants? There may be a variety of the participants from the experienced expert to the novice end user.
- The control of the deletion or the modification of input is necessary to prevent to interfere with other participants' knowledge. This difficulty is common to groupware.

To solve these difficulties, we designed GRAPE on the following principles. These principles are reflected to the interfaces and the knowledge merge methods.

- The merging methods are designed as simple as possible.

  If the knowledge consists of numerical values, the average is used to merge the knowledge. It is the simplest method and easy to change the importance of the participants.

  If the knowledge consists of the names, the concatenation of the names are used as shared knowledge.

In case the knowledge cannot to be concatenated nor averaged, for example the tree structure, we developed original methods to merge the knowledge[11]. For example, to merge the clustering tree of the candidates, we developed the fuzzy clustering method. It transforms the tree into an averagable matrix form. Then it makes an average of matrices and transforms them back into tree form.

- The knowledge from every participant is regarded equally.

  This principle is the same as in brain storming.

- The modification or deletion of knowledge is only permitted by the participant who input it.

  This principle is difficult in some situations. For example, even the experienced expert cannot correct the knowledge inputed by the other participants.

  However, it is not a problem in general because the participants do not have to input unsure knowledge and every kind of knowledge is checked with consistency locally using the system engineering methods.

- The participants share the knowledge as much as possible, by showing every knowledge any time.

  This principle makes the WYSIWIS interface.

- GRAPE permits the incomplete knowledge. In other words, the participants inputs the knowledge so long as they know well. The missing of the knowledge is completed by the knowledge from the other participants and default of knowledge.

2.4 Outline of GRAPE

This section explains the outline of GRAPE with the module composition. GRAPE system consists of 3 modules and the second module consists of 5 sub-modules (Table 1). The execution progress on this sequence.

In the initialization module, the chairperson who is called the "coordinator" and the connection with the hosts of the participants are decided. In the current implementation, the first user who started the system becomes a coordinator. The coordinator is the same as the participants except for the role to confirm the end of each step.

The knowledge acquisition module consists of five sub-modules: Candidates acquisition, Candidates structuring, Attributes acquisition, Attributes structuring, and Classes evaluation.

In the candidates acquisition sub-module, the system urges all the participants including the coordinator to input the names of the candidates of the solution to the problem.

In the candidates structurizing sub-module, the system urges the participants to input knowledge to structure the candidates. As a result, a tree structure is acquired, in which the candidates exist as a leaf. To make this tree, each participant inputs the degree of the similarity between candidates with the number from 0.0 to 1.0. The higher similarity degree connects the candidates by the branch the nearest leaf. The similarity degree 1.0 means these candidates are equivalent and 0.0 means these are irrelevant.

Module name	Contents and used methods
1 Initialization module	Decision of the coordinator and the other participants
2 Knowledge acquisition module	
2.1 Candidates acquisition	Acquisition of the candidates using WYSIWIS interface
2.2 Candidates structuring	Acquisition of the similarity value between each candidate, structuring the candidates using Fuzzy Clustering, and acquisition of the names of the clusters in the structure
2.3 Attributes acquisition	Acquisition of the attributes distinguishing the clusters and the candidates using the elicitation method of PCP
2.4 Attributes structuring	Acquisition of the dependency between each attribute and structuring the attributes using Extended ISM
2.5 Classes evaluation	Evaluation of the importance between the attributes and the evaluation between the candidates with each attribute using AHP
3 Calculation result	Integration the evaluation of the candidates from the results of AHP

Table 1: System flow of GRAPE

Then the names of the cluster, which exist on the branch, is acquired from the participants.

In the attribute acquisition sub-module, the names of the attributes are acquired. For this acquisition, the interview technique is used to elicit attributes in PCP[8]. The acquired names of the attributes are used as the criterion to evaluate the candidates and the clusters of the candidates in the later classes evaluation sub-module.

In the attribute structuring sub-module, the system urges the participants to input knowledge to structure the attributes obtained in the previous sub-module. For this purpose, the participants input the dependency between the attributes. A tree structure is obtained by using Extended ISM (Interpretive Structural Modeling)[9]. In this tree structure, the depending attribute is placed near the root and the depended attribute is placed near the leaf. The mutually depending attributes are considered the equivalent attributes and just one of them are used. The obtained tree structure is used as a tree structure of AHP.

In the classes evaluation sub-module, the evaluation of each branch is performed by using AHP (Analytic Hierarchy Process)[10]. To evaluate each branch, the system urges input of pairwise comparisons between the candidates (or the clusters of the candidates) about each attribute. The system also urges input of pairwise comparisons about the importance between the attributes. These comparisons are urged at each branch of the clustering tree of the candidates.

In the calculation result module, the evaluations at every branch are integrated and the results of the evaluations of each candidate are displayed to the participants.

3 Detail of the methods used to acquire or merge knowledge

GRAPE uses various methods for system engineering. These methods are modified more or less to suit for groupware. Some methods were originally developed for this purpose. We will show these methods in the same sequence of the acquisition. The methods are as follows:

- Fuzzy Clustering: to make a clustering tree from the similarity degrees,
- PCP (Personal Construct Psychology): to elicit the attributes to distinguish the candidates,
- Extended ISM (Interpretive Structural Modeling): to describe the dependencies among the attributes, and
- AHP (Analytic Hierarchy Process): to elicit the importance between each attribute and to elicit evaluation values about each attribute between each candidate. The Harker's method are also used to reduce the amount of input during AHP. This method bring out the knowledge from the incomplete AHP matrix.

3.1 Fuzzy Clustering

Fuzzy Clustering is used to make a clustering tree from the similarity degrees.

In tree structure, the candidates exist on the leaf and the candidates which have the higher similarity degrees are connected by the branch at the nearer to the leaf. To make this tree, each participant inputs the degree of the similarity between the candidates with the number from 0.0 to 1.0. The similarity degree 1.0 means these candidates are equivalent and 0.0 means these are irrelevant.

These degrees make an matrix. If this matrix is the fuzzy similarity matrix[12], then the matrix is equivalent to the tree structure and these can be transformed each other, as in figure 1. The fuzzy similarity matrix is a matrix where: • every element is between 0 to 1, • it is symmetric, • the diagonal elements are all 1, and • it transitive: A transitive similarity matrix $\{a_{ij}\}$ satisfies $^{\forall}ij\ (a_{ij} \geq \max_k(\min(a_{ik}, a_{kj})))$.

GRAPE fills the elements determined by transitivity of the matrix automatically. It reduces the number of inputs to the number of the candidates. GRAPE shows the tree structure immediately as the participants input the similarity, by supposing the undetermined elements as 0.

We developed a new merge algorithm which is named Fuzzy Clustering[11]. The Fuzzy Clustering algorithm has 2 steps. First, an average matrix is computed using arithmetic

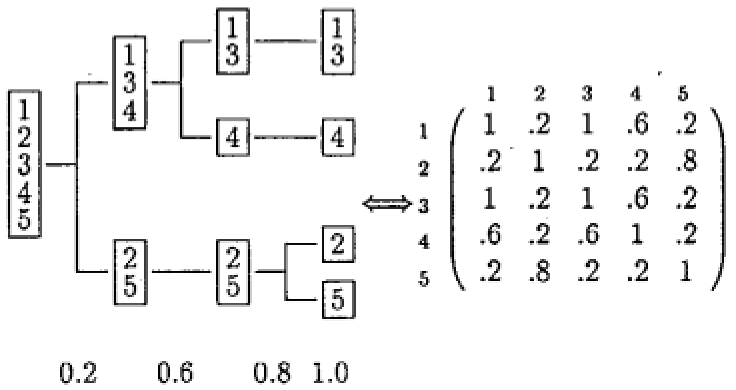

Figure 1: Transformation between tree structure and fuzzy similarity matrix

mean, then it is changed to be transitive because the average matrix is not necessary to satisfy transitivity. To make the matrix to be transitive, the matrix is copied to new matrix from the bigger elements. And during copy, the elements decided by transitivity are filled and the lower elements in the averaged matrix are discarded.

3.2 Other methods

PCP(Personal Construct Psychology[8]) is used to elicit the name of attributes to evaluate the candidates and the classes. PCP consists of various methods to construct the evaluation structures. We only used the elicitation methods to elicit the attributes. This method is a kind of the interview technique.

To make the elicitation fit to groupware, the WYSIWIS interface is used as well as the acquisition of the candidates. And the names of the inputed the attributes are shown immediately to the window of every participant.

Extended ISM (Interpretive Structural Modeling) is used to describe and to arrange the attributes. This extension is worked by one of the authors[9]. The arrangement of the attributes deletes the duplicated attributes and constructs a tree structure of the attributes based on dependency. In this tree structure, the depending attribute is places near the root and the depended attribute places near the leaf. The attributes independent on the other attributes place just under the root. The attributes mutually depended are considered equivalent and just one of them are used.

The acquisition of the dependencies uses the WYSIWIS interface as well as the acquisition of the candidates. To make merged knowledge, GRAPE gathers the dependencies from all the participants and extracts a skeleton tree from these dependencies.

AHP (Analytic Hierarchy Process)[10] is used to evaluate the candidates and the clusters of the candidates from the viewpoint of each attribute. As a result, preferences between the candidates and the clusters are obtained.

The Harker's Method[13] can be used to reduce the amount of the inputs. By using

Harker's method, the evaluation values can be calculated from an incomplete matrix of the pairwise comparisons.

To make the evaluation fit to groupware, GRAPE makes an average matrix of all the matrices from the participants before calculating an eigenvector. The average matrix is computed using geometric mean. Then the average preference is calculated from the average matrix in the same way to AHP. Each participant can see both private and average preference vector.

4 Demonstrations of GRAPE

We will show the display images of the example to demonstrate the sequence of the execution. The problem of this example is to chose the best computer for groupware.

First, one coordinator starts the GRAPE system, and inputs the name of the problem. In this demonstration, it is "Groupware" Then the coordinator inputs the names of the hosts of the participants and these hosts are connected to the host of the coordinator. After the connection, the coordinator has the same role as the other participants except for the role to confirm whether the knowledge acquired in the process of each sub-module is complete. The decision which progresses to the following execution process will be done if it becomes complete.

Then, it creates three windows for each host: • Message window, • Text input window, and • Tree output window. The message window is displayed at the top of the window system by default. The text input window and the tree output window are displayed at the bottom left and at the top right of the window system respectively (Figure 2). A waiting indicator is at the top left of the window system. This indicates the participants who have finished work and are waiting by showing the host name in a white letter on a black background. At first all of the hosts' names are shown in a normal black letter on a white background.

Now, it begins the first candidates acquisition sub-module . Each participant including the coordinator inputs the candidates of the solution of the problem. For this example, "Macintosh," "SUN," and "PSI" is inputed as in the figure 3. All the candidates are shown in the tree output window as soon as inputed and every user understand what candidates are rising in one glance. This mechanism is called WYSIWIS (What You See Is What I See.) This interface gives each user the feeling that they are cooperating each other. It also stimulates the idea of the respective participant by showing the idea inputed by the other participants. This effect is similar to one in the brain storming.

After the confirmation of the coordinator, it proceeds to the candidates structuring sub-module. After here we do not notice the point of the confirmation of the coordinator to avoid to be persistent. There are several points of the confirmation in and after every sub-module.

To structuring the candidates, each participants inputs the similarity values between the candidates. During this sub-module, the similarity matrix is displayed at the left of the window system. The participant can see the tree structure in the tree output window and can see the tree structures of the other participants and the average tree structures if necessary (Figure 4). These optional tree window is shown at the bottom of the window

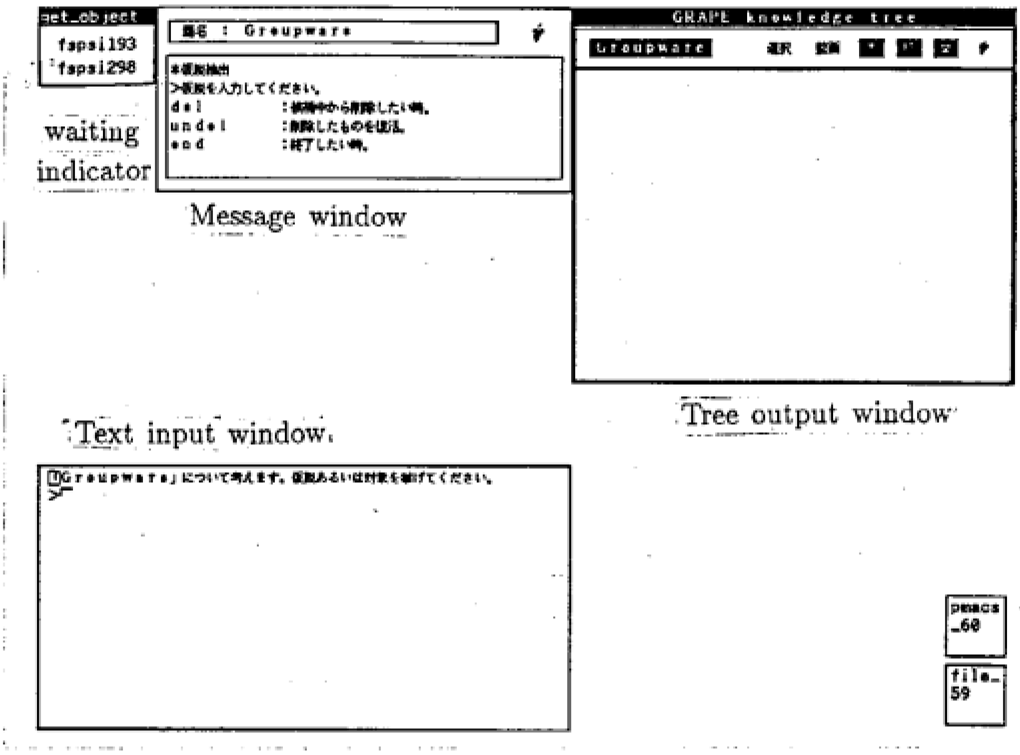

Figure 2: Initial window settings

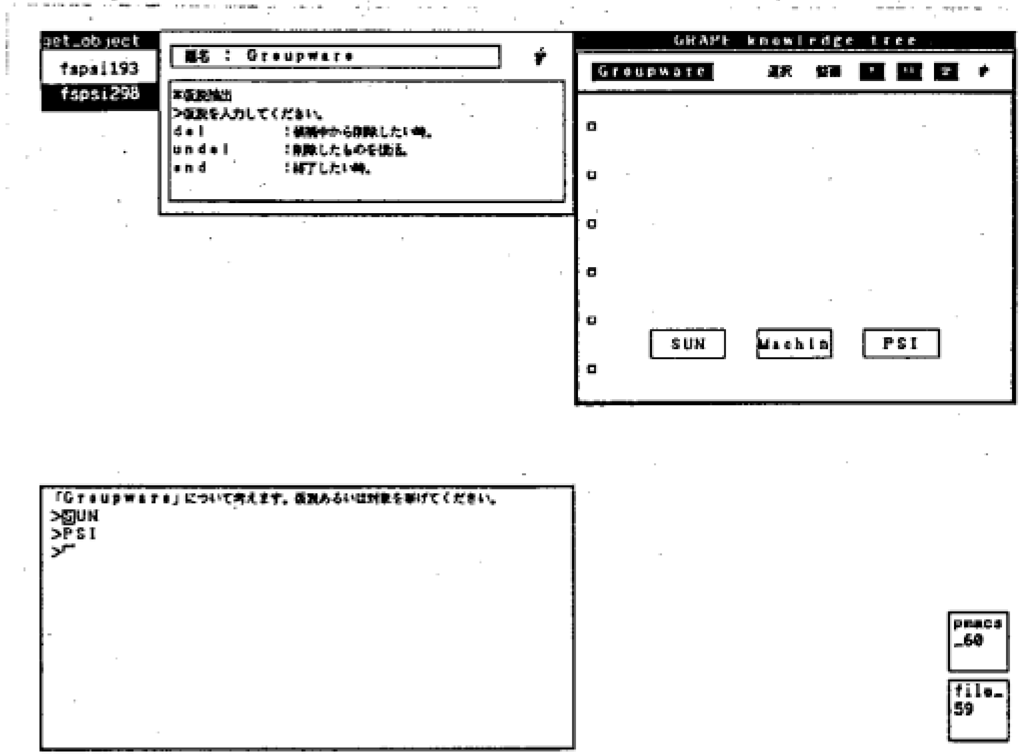

Figure 3: Input the candidates using WYSIWIS interface

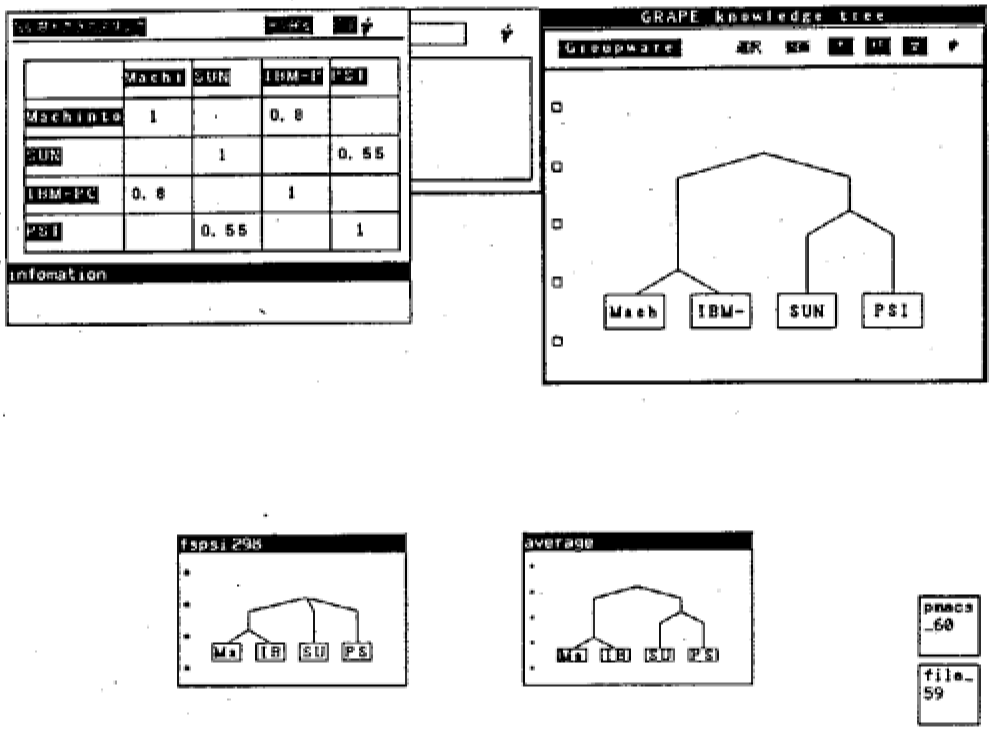

Figure 4: Clustering the candidates based on the similarities

system. After the system merges the trees by using the Fuzzy Clustering method, Each participant inputs the names of the clusters at the branches of the tree.

Then, it proceeds to the next sub-module of attributes acquisition. Each participant inputs the names of the attributes to evaluate the candidates and the cluster of the candidates. For this elicitation, PCP is used. Then, these attributes are structured by using Extended ISM. Figure 5 shows the display after Extended ISM. The window named "detail window" at the center displays the detail of a branch: the cluster name, the attribute names, the dependency between the attributes, and the tree structure obtained by Extended ISM.

Now, there is enough knowledge to begin the evaluation of the candidates. Then it proceeds to the last classes evaluation sub-module. In this sub-module, AHP is performed at every branch in sequence. Each AHP process has a small tree obtained by Extended ISM, and each branch of the small tree has a matrix for the pairwise comparisons. Figure 6 shows the display of the comparison. There are two windows displaying the trees; the right one is the tree obtained by Fuzzy Clustering and the other is the small tree obtained by Extended ISM for the branch indicated in the Fuzzy Clustering tree.

Then the knowledge acquisition module comes to the end and it proceeds to the calculation result module. This module integrates the preference vector on each branch obtained at the last sub-module and shows the result preference of the candidates to the tree output window (Figure 7).

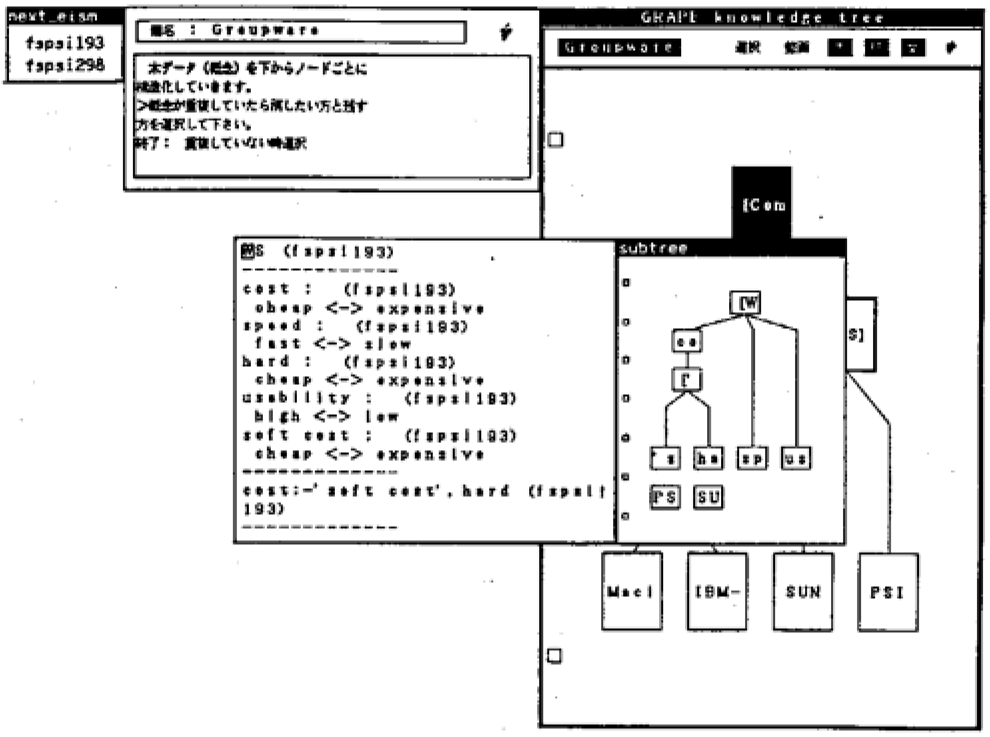

Figure 5: The dependencies between attributes; shown in the detail window

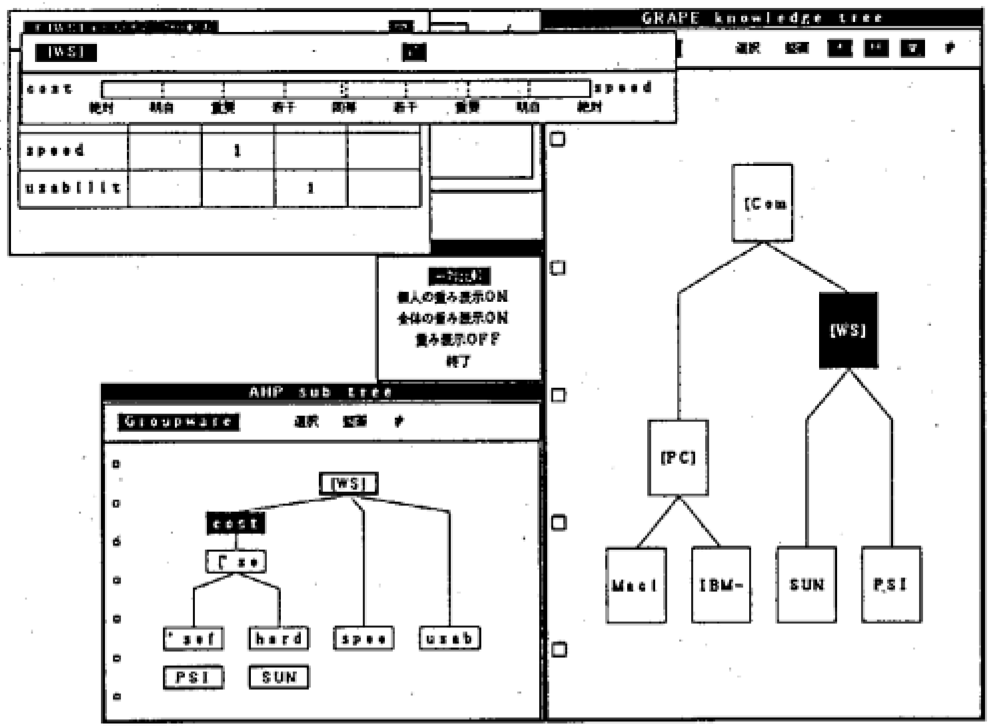

Figure 6: The importance between the attributes

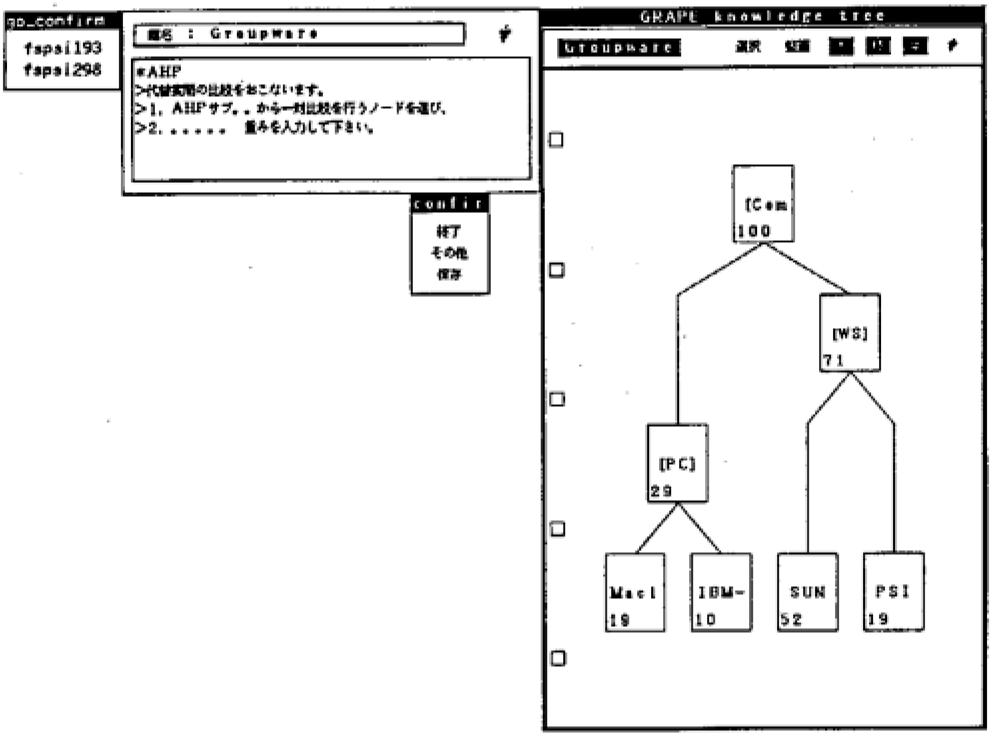

Figure 7: The result preferences of the candidates

5 Extension for the planning type system using the constraint logical language: GCL

We limited GRAPE as a knowledge acquisition tool for the classification-choice type knowledge-based system. It is because the choice of the solution is the common difficulty of any kind of system. However, in a system for the planning, the problem of how to generate the candidates is as difficult as how to select the solution from the candidates. It is because there are huge numbers of solutions. To find some candidates needs so much computation if the candidates are generated in a simple way. This is why the ordinary knowledge acquisition tool for the planning type system acquires the way how to generate the candidates as well as the way how to choose a solution. This method sometimes does not work especially for the knowledge acquisition for a rapid prototype system. It is because the way how to find the solution is closely depending on the way how to evaluate the solution, and the changes of the evaluation method are often needed to make a rapid prototype system.

Now, a kind of logical language named GCL was designed to generate solutions efficiently[2]. GCL is designed to solve the constraints, which is described using inequality, non-equal relation, and linear inequality, and to produce the only solutions satisfying these constraints. GCL also handles the non-linear evaluation function, and search the solutions using the best-first search method. Adding to them, a special constraint is preserved, by which users can describe permutation directly because the permutation is the reason of the huge computation. The explicit declaration of permutation helps to system to optimize the generation of the solutions.

To use GCL, GRAPE can be extended to acquire knowledge for the planning type system. The extended GRAPE may do the following processes: First, it acquires constraints of the valid solutions and evaluation function of the solution, then GCL generate candidates of solutions using these constraints and evaluation function, last, current GRAPE is applied to these candidates and the evaluation between the candidates are integrated into the evaluation function. These process are repeated until the participants satisfy.

6 Conclusion

GRAPE is a knowledge acquisition support groupware and acquires knowledge for the classification-choice type knowledge-based system. GRAPE is similar to ETS and YUAI from the point of view of knowledge acquisition system, but it incorporates many features as groupware. To compare with ETS, GRAPE can evaluate the knowledge using AHP. The actual choice from the candidates is equivalent to a test. The participants can choose the other candidates by changing the importance preference vector at any branches. These retry to the choice takes similar effects to the sensitivity analysis.

GRAPE is similar to GDSS (Group Decision Support System) in the view that it uses the decision support methods. Unlike the GDSS, the contents of the agreement are not important. GRAPE does not has any facility to negotiation. It is because GRAPE is a knowledge acquisition tool. The knowledge is basically common to all the participants, so these are merged into a shared knowledge without negotiation.

While GRAPE intended to avoid the backtracking, the participants often would like to backtrack and add some knowledge. We find there are two typical backtracking: One is the backtracking from Fuzzy Clustering to adding the candidates, and the other is the backtracking from Extended ISM to adding the attributes. Both kinds of the backtracking are invoked by the structuring of the knowledge. It may be because the tree structure makes clear view about the candidates. It is expected that the integrated methods to acquire both items and its structures from multiple users incrementally.

GRAPE is designed to reduce the amount of inputs, but the amount of inputs increased because the participants stimulates each other and tend to input the knowledge concern with the knowledge which the other participants have inputed. In other words, the groupware facilities tends to make high quality knowledge. It is the effect often said "Two heads are better than one."

We have shown an extension for planning type system. There is another extension of GRAPE. It is to extend GRAPE to acquire knowledge for the diagnosis type system.

To extend GRAPE for the diagnosis type system, GRAPE needs to have the mechanisms to evaluate the candidates using objective values rather than subjective values. The system also needs to be able to describe the test methods or the test expressions of objective values.

Integrating the extension for the planning type system and diagnosis type system will make GRAPE more powerful.

Acknowledgment

This research was a part of work charged from ICOT. We thank our colleagues Dr. Shintani[14] and Dr. Hiraishi[15] for permission to use the modules concerning AHP and tree display windows. We thank Dr. H. E. Peterson and Ms. M. Fujita for proofreading this paper.

References

[1] M. Stefik, G. Foster, D. G. Bobrow, K. Kahn, S. Lanning, and L. Suchman. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *CACM*, 30(1):32–47, January 1987.11987.

[2] H. Ueda and S. Kunifuji. GCL: Parallel constraint logical language for planning problem. Technical Report Comp91-64, IEICE, 1991. (In Japanese).

[3] J. Boose. Knowledge acquisition tools, methods, and mediating representations. In *Proc. of JKAW'90*, 1990.

[4] John H. Boose. A knowledge acquisition program for expert systems base on personal construct psychology. *International Journal of Man-Machine Studies*, 23:495–525, 1985.

[5] John H. Boose and M Bradshow. Expertise transfer and complex problems: using AQUINAS as a knowledge-acquisition workbench for knowledge-based systems. *International Journal of Man-Machine Studies*, 26:3–28, 1987.

[6] S. Sawai, et al. YUAI: a knowledge acquisition system. In *Proc. of 38th National Conf. of Information Processing Society of Japan*, pages 127–128, 1989. (in Japanese).

[7] J. R. Quinlan. *Learning efficient classification procedures and their application to chess endgames*, volume 1. Tioga Pub., 1983.

[8] G. A. Kelly. *The Psychology of Personal Constructs*. Norton, 1955.

[9] S. Kunifuji and H. Ueda, et al. The methods to strucrize the attributes for the knowledge acquisition support groupware : GRAPE. In *Proc. of the The 11th Symposium on the intelligent system*, March 1990.31990. (In Japanese).

[10] Thomas L. Saaty. *The Analytic Hierarchy Process*. British Library Cataloguing in Publication Data. McGraw-Hill, 1980.

[11] H. Ueda and S. Kunifuji, et al. The methods to structurize the hypothesis for the knowledge acquisition support groupware : GRAPE. In *Proc. of the The 11th Symposium on the intelligent system*, March 1990.31990. (In Japanese).

[12] L. A. Zadeh. Similarity relations and fuzzy ordering. *Inform. Sciences*, 3:177–200, 1971.

[13] P. T. Harker and L. V. Vargas. The theory of ratio scale estimation: Saaty's ahp. *Manegement Science*, 33:1383–1403, 1987.

[14] T. Shintani. A rule based consistency maintenance for subjective judgments. In *Proceedings of International Conference on Information Technology, Commemorating the 30th Anniversary of The Information Processing Society of Japan*, volume 2, pages 179–186, 1990.

[15] M. Toda, K. Hiraishi, T. Shintani, and Y. Katayama. Information structuring and its implementations on a research decision support system. *Decision Support Systems*, 7(2):169–184, 1991.